INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Reducing the number of binary splits, in Decision Tree Induction, by means of an Hierarchical Classification

Israël-César Lerman, Joaquim F. Pinto Da Costa

No 3312

Décembre 1997

THÈME 3

ISSN 0249-6399

Reducing the number of binary splits, in Decision Tree Induction, by means of an Hierarchical Classification

Israël-César Lerman*, Joaquim F. Pinto Da Costa†

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet REPCO

Rapport de recherche no 3312 — Décembre 1997 — 38 pages

Abstract: The main problem considered in this paper consists of binarizing categorical (nominal) attributes having a very large number of values (20^4 in our application). Few number of relevant binary attributes are gathered from each initial attribute. The significant idea consists in grouping the values of an attribute by means of an hierarchical classification method. The similarity between values is associated with the classes to be predicted. The solution that we propose is independant of the number of these classes and can be applied to various situations. A specific use of the obtained classification tree reduces very significantly the number of binary splits of the attribute value set that have to be retained. In fact and for complexity reasons, the hierarchical classification method is combined with formal decomposition and recomposition of the attribute value set. The ARCADE method that we have set up is mainly a powerful hybridation of the celebrated CART method, by our above outlined reduction method. The application of ARCADE, to the protein secondary structure prediction problem, proves the validity of our approach.

Key-words: Decision trees, binarization, complexity reduction, categorical attributes, hierarchical classification.

(Résumé : tsvp)

* IRISA-INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cédex, lerman@irisa.fr ; tél : +33 (0)2 99 84 72 86 ; fax : +33 (0)2 99 84 71 71

† Departamento de Matemática Aplicada ; LIACC ; Univ. do Porto, Rua das Taipas, 135, 4050 Porto, Portugal, jpcosta ncc.up.pt ; tel :+351 2 6001672 ; fax :+351 2 2004109

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : 02 99 84 71 00 - International : +33 2 99 84 71 00
Télécopie : 02 99 84 71 71 - International : +33 2 99 84 71 71

Réduction du choix des attributs binaires dans un arbre de décision au moyen d'une méthode de classification hiérarchique

Résumé : Le problème principal traité dans ce travail est celui de la récolte d'un bon choix d'attributs binaires, en "petit" nombre, à partir d'attributs qualitatifs ayant un très grand nombre de valeurs (20^4 dans notre application). L'idée centrale consiste dans le groupement des valeurs de l'attribut prédictif qualitatif initial, au moyen d'une méthode de classification hiérarchique. La similarité entre valeurs dépend précisément des classes à prédire. Il est d'importance de noter que la solution que nous proposons est indépendante du nombre de ces classes et peut être appliquée dans de nombreuses situations. La structure ultramétrique de l'arbre des classifications est mise à profit pour une réduction vertigineuse du nombre de coupures binaires à retenir de l'ensemble des valeurs d'un attribut prédictif initial. En fait, pour des raisons de consistance statistique et de complexité, la méthode de classification hiérarchique est combinée avec une décomposition formelle et recomposition après classification de l'ensemble des valeurs de l'attribut. La méthode ARCADE que nous avons bâtie est principalement une puissante hybridation de la célèbre méthode CART, au moyen de notre méthode de réduction. L'application d'ARCADE au problème de la prédiction de la structure secondaire d'une protéine, établit la validité de notre approche.

Mots-clé : Arbres de décision, binarization, complexité, attributs qualitatifs, classification hiérarchique.

1 Introduction

Binary decision trees are a representative logic tool of discrimination of concepts. The methodological and inferential problems relating their elaboration concern two domains : qualitative data analysis and machine learning ([Breiman & al., 1984],[Quinlan, 1986],[Nakhaeizadeh, 1994],[Taylor, 1994]).

Their construction, which depends on a set $\mathcal{A}$ of binary description attributes, is based on a learning set $\mathcal{O}$. The concept (or class) to be recognized is expressed as an attribute c taking K different values, where K is not necessarily very small.

Predictive descriptive attributes are seldomly binary at the beginning. Although binarization is often an easy task (for ordered variables or nominal with a small number of values), in our work this is not at all the case. Let $\mathcal{E}(v)$ be the value set of an attribute v. A fundamental aspect of concern has been the research of how to construct two-class partitions of $\mathcal{E}(v)$, in order to obtain the binary attributes associated with v. This construction must depend on the structure (or semantic) of $\mathcal{E}(v)$. Lately, the concern has mainly been on finding ways to discretize the continuous attributes ([Krzanowski, 1975], [Breiman & al., 1984], [Van de Merckt, 1993], [Heath & al., 1993], [Müller & Wysotzki, 1994]). In the methodological developments that we consider in this article, each of the predictive attributes v is qualitative nominal (categorical) and takes a very large number of different values. In other words, $\mathcal{L} = \mathcal{L}(v) = card[\mathcal{E}(v)]$ is large and in our data it can reach 20^4 values for each predictive attribute v. Then, even a complexity $O[L.log(L)]$, as reached in the celebrated program CART [Breiman & al., 1984] for $K = 2$ classes, cannot be acceptable in our case. Nevertheless, as we shall see, this is not the main reason why CART is not applicable to our data.

Our method, ARCADE, by grouping the values of each predictive attribute v into clusters, allows the definition of another qualitative attribute with much fewer values. For example, let us consider an attribute v with 8 values $m_1, m_2, \ldots, m_8$. By grouping these values into three clusters, $\{m_1, m_5, m_8\}, \{m_2, m_3\}, \{m_4, m_6, m_7\}$, we can define a new nominal (categorical) attribute with three values. And then, we can associate with it three binary attributes, each attribute being defined by the union of the values grouped in the corresponding cluster. In order to make signifiant these latter attributes, the values of the attribute v grouped in the same cluster have to be near with respect to their statistical behaviour on the classes to be predicted. In these conditions, a contingency table which crosses the predictive attribute v, with the attribute c to be predicted, is established. The application of an automatic classification method to the rows of this contingency table, which represent the values of v, allows the definition of a new categorical attribulte $\bar{v}$. By denoting $L(\bar{v})$ the number of its categories, the number of binary splittings of $\bar{v}$, that have to be considered in the decision tree construction, is $(2^{L(\bar{v})-1} - 1)$. This number is equal to 3 in the above example. It becomes too large in case where $L(\bar{v})$ is not small enough. And, instead of a non hierarchical clustering approach, we apply a hierarchical classification method to the rows

of the contingency table. The binary splittings which are retained are then associated with the leaves or nodes of the classification tree on the value set of the new categorical variable $\bar{v}$. This technique reduces very considerably the complexity of finding "good" predictive attributes.

However, in case where the number of rows is too large (20^4 in our case), the direct utilization of a classification method on the row set of the contingency table, becomes - for complexity and statistical reasons - neither tractable nor significant. To resolve this problem we propose a decomposition and agregative recomposition method on the value set of the attribute v. More precisely, in our application problem (see section 2), v is factorized into a pair (v^1, v^2) of attributes. Each of the latter attributes v^1 and v^2 gives rise to a contingency table of which the order of the row size is $\sqrt{L(v)}$. Therefore, by applying a classification method to both contingency tables, the pair of attributes (v^1, v^2) is reduced to a pair of synthetic attributes that we denote $(\bar{v}^1, \bar{v}^2)$. The final contingency table that we consider, crosses the value set of $(\bar{v}^1, \bar{v}^2)$ with the value set of the attribute to be predicted, c.

In the construction of a binary decision tree, at each node of the latter, one must choose a binary attribute to split that node into two descendent nodes. This choice is based on a measure of association, or cœfficient, between the binary attribute and the concept (attribute) to be predicted. As has already been demonstrated ([Buntine & Niblett, 1992],[Liu & White, 1994],[Lerman & Costa, 1996],[Costa, 1996]), a "good cœfficient" must be used. The method ARCADE presents a large choice of cœfficients, including the Gini cœfficient (used in CART), the information of Shannon, and the Lerman cœfficients. [Costa, 1996] has demonstrated that the use of the Gini cœfficient in decision trees is equivalent to the use of the cœfficient of Haldane (1940). The fact that the cœfficient of Haldane has been conceived to predict, explains therefore the good results that are usually obtained with the Gini cœfficient.

We will give now a bried description of our data :

1.1 Prediction of protein secondary structure

The primary structure of a protein is formally a word of which the letters are taken from an alphabet of 20 letters (corresponding to the 20 amino acids A,C, ...,Y). The length of such a word can reach many hundred letters. The associated secondary structure of a protein can also be formally defined as a word of the same length, but the corresponding alphabet contains three letters, E, H and X, which are the names of the three concepts to be discriminated (H corresponds to an helix, E to a strand and X stands for coil) :

...	T	T	C	C	P	S	I	V	A	R	S	...
...	*E*	*E*	*E*	*E*	*X*	*X*	*H*	*H*	*H*	*H*	*H*	...

This prediction problem has been attacked by many researchers over the last two decades, although we are not aware of anyone using decision trees. For instance, more recently, [Qian

& Sejnowsky, 1988], have used neural networks for treating this problem ; [Rost & Sander, 1993] have also used neural networks, and they have reached, for the first time, 70.1% of correct classifications. [Cost & Salzberg, 1993] have used nearest-neighbor methods ; [Zhang, Mesirov & Waltz, 1993] have used a combined neural-net/nearest-neighbor/Bayesian approach ; [Solovyev & Salamov, 1994] have used four linear discriminant functions for predicting secondary structure segments, instead of predicting one residue at a time.

Although the number of known primary sequences increases extremely fast, the same is not true for the secondary structure, and so, there is a widening number of protein sequences whose secondary structure needs to be predicted. This is not a difficult problem when there are proteins, of known secondary structure, which are homologous to the protein being predicted ; but, for 85% of the new proteins, there does not exist homolog proteins ; and the problem becomes very difficult. Most of the existing prediction methods try, for a *description* of each letter of the first word, to predict the corresponding letter of the second word. Yet, it is crucial to find a pertinent description of the data for this difficult recognition problem. Indeed, the prediction does not only depend on the letter of the first word corresponding to the letter of the second word that has to be predicted, but also on its neighbourhood.

The predictive attributes that we have adopted will be defined in section 2. We will describe in this section the different stages of the application of the ARCADE method to this difficult prediction problem. As a matter of fact, it is by resolving this problem that we have set up the concerned methodology. This has been validated by the obtained results.

More autonomous and formal definition of ARCADE will be expressed in section 3. Comparison with CART method in terms of computational complexities, will also be evocated in this section.

In order to agregate rows of a contingency table, according to a hierarchical scheme, the AVL ("Analyse de la Vraisemblance des Liens") hierarchical classification method is employed. This method is extremely general with respect to the logical or mathematical structure of the data to be organized, according to their resemblances. This approach will be outlined in section 4.

Finally section 5 will be devoted to conclusion and perspectives.

2 Binary attributes used in ARCADE for the prediction of protein secondary structure

2.1 The initial descriptive attributes

As said above, the description of the position to be predicted, must take into account all its neighbourhood in the protein sequence (see § 1.0.1 above). For this purpose, we begin by considering a window having an eleven size $2p+1$ (p : integer), centered on the position to be predicted. Different values of p have been tried and the best prediction results have been obtained for the experimental value $p=5$. If we designate by y the position of the protein whose secondary structure we want to predict then, for a window of size eleven, the first descriptive attributes used by us are $v^1, v^2, \ldots, v^{11}$:

	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9	v^{10}	v^{11}	
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	
	y_{-5}	y_{-4}	y_{-3}	y_{-2}	y_{-1}	y	y_{+1}	y_{+2}	y_{+3}	y_{+4}	y_{+5}	
...	T	T	C	C	P	S	I	V	A	R	S	...
						?						

These 11 nominal attributes, which take 20 different values (because there are 20 different amino acids), are natural to consider ; nevertheless, since the order between the different amino acids is most important for prediction, we have decided to generalize and to make more accurate the above description.

For this purpose we consider a "Multiplication" formal operation of these elementary attributes, in a consecutive manner. More precisely, by associating two by two, in an adjacent way, the preceding eleven attributes, we define $11-2+1=10$ composite new attributes. Such set of attributes is determined by a subwindow having two positions and sliding along the whole window of size eleven. Namely, these attributes are

$$(v^1, v^2), (v^2, v^3), (v^3, v^4), (v^4, v^5), (v^5, v^6), (v^6, v^7), (v^7, v^8), (v^8, v^9), (v^9, v^{10}) \text{ and } (v^{10}, v^{11}).$$

Each attribute can be expressed as a word with two letters. Its value set is defined by the cartesian square $\mathcal{A} \times \mathcal{A}$, where $\mathcal{A} = \{A, C, \ldots, Y\}$ and represents the 20 amino acids. Thus, the size of this value set is 400.

More generally, one may consider attribute words with a number of letters varying between one and eleven. However note that the possible number of values of a word with l letters, is 20^l. And this number becomes extremely large if l is not small enough. Thus, for $l=4$, $20^l = 160,000$. To fix ideas consider this particular value of l, in order to make clear the definition of the predictive descriptive attributes.

Definition. A descriptive predictive attribute is a word of four consecutive letters having a given position in the sliding window of size eleven.

Therefore, there are in all $11 - 4 + 1 = 8$ attributes, and for each instanciation of the sliding window, we get one observation of each of the eight attributes. On the other hand, we associate with these respective values the observation of the secondary structure, at the center of the window.

Now, consider one specific attribute (to fix ideas, you may suppose that it concerns the word occupying the positions 5,6,7 and 8, of the window) and assume that we have in the learning set 100 protein sequences with a common lenght, 310. In these conditions, we have in all $300 \times 100 = 30,000$ observations. These can take position in the contingency tables which crosses the specified attribute with the concept attribute. Each observation adds one unity in the concerned entry of this matrix, of which the size is $20^4 \times 3 = 480,000$.

As expressed above (see Introduction) the general idea consists in clustering the rows of the contingency table in order to reduce the value set of the descriptive attribute. But it is irrelevant to proceed directly. Because, this table is too sparse to allow significant classification of its rows. Moreover, there are at most 30,000 rows which are non empty and then, the whole 160,000 possible values of the specified attribute cannot be represented. Otherwise, for computing complexity reasons, there is no clear solution for handling so large data set, with known clustering algorithms.

2.2 Building the binary attributes used in ARCADE

In the introduction, the solution that we have adopted was briefly described. It consists in first, "Factorizing" the 4-letter word attribute w, into an ordered pair of 2-letter words attributes (w^1, w^2), where $w = (w^1, w^2)$. w^1 and w^2 are factors of the factorization of w. A classification reduction procedure on the value set of w^1 (resp. w^2) leads to the statistical reduction of the value set of w. More details of this construction are going to be given.

Let us assume description with a 2-letter word. In this case, the size of the contingency table is $20^2 \times 3 = 1,200$. 30,000 observations are distributed over 1,200 entries of this table which then becomes statistically consistent.

Consequently, in order to agregate the set value of the above attribute w, we begin by respectively agregating the value sets of w^1 and w^2. For this purpose we substitute to the initial contingency table crossing w and c, an ordered pair of contingency tables crossing w^1 and c, respectively, w^2 and c. This corresponds statistically to replace the joint distribution of (w^1, w^2) on c, by the product of the marginal distributions of w^1 and of w^2, on c.

w \ c	X	H	E
1(AAAA)			
2(AAAC)			
$\vdots$			
20^4(YYYY)			

FIG. 1 - Initial contingency table.

w^1 \ c	X	H	E
1(AA)			
2(AC)			
$\vdots$			
20^2(YY)			

w^2 \ c	X	H	E
1(AA)			
2(AC)			
20^2(YY)			

FIG. 2 - Contingency tables for the factorized attributes.

We have now to employ a classification method on each row set of both contingency tables, independently. Obviously, the similarity between rows is evaluated differently according to the concerned contingency table. Let us designate by $\{W_1^1, \ldots, W_i^1, \ldots, W_l^1\}$ and $\{W_1^2, \ldots, W_j^2, \ldots, W_m^2\}$ the obtained classifications on, respectively, the value sets of w_f^1 and w_f^2. The first stage clustering of the value set of w is obtained by means of the cross product

$$\{W_1^1, \ldots, W_i^1, \ldots, W_l^1\} \times \{W_1^2, \ldots, W_j^2, \ldots, W_m^2\}.$$

More precisely, for a given (i, j), $1 \leq i \leq l$, $1 \leq j \leq m$, $W_i^1 \times W_j^2$ is considered as a single value of a reduced version $\bar{w}$ of w. Thus, all the 4-letter words belonging to $W_i^1 \times W_j^2$ are considered as equivalent.

For comparable accuracy in both classifications, l and m must be more or less equals. On the other hand, new reduction procedure of the value set of $\bar{w}$ is still necessary on the basis of the associated $(\bar{w}, c)$ contingency table. Therefore, for statistical and computing reasons, the size of $l \times m$ cannot be too large. By choosing this size about one thousand, the ratio between the number of observations and the number of the entries of the contingency table is around 3, for the above illustrated case of 30,000 observations. Pratically, we have considered $l = m = 31$; and then, $l \times m = 961$.

$\bar{w}$ \ c	X	H	E
$W_1^1 \times W_1^2$			
$W_1^1 \times W_2^2$			
$\vdots$			
$W_1^1 \times W_{31}^2$			
$W_2^1 \times W_1^2$			
$W_2^1 \times W_2^2$			
$\vdots$			
$W_{31}^1 \times W_{31}^2$			

FIG. 3 - Contingency table for the reduced attribute.

In our approach the partition $\{W_1^1, W_2^1, \ldots, W_{31}^1\}$ (resp. $\{W_1^2, W_2^2, \ldots, W_{31}^2\}$) is gathered at a given level of the classification tree built by the hierarchical classification method $AVL_{0.5}$ (see sections 4 and 5) applied to the rows of the contingency table (w^1, c) [resp. (w^2, c)] (see above). Considering $\{W_1^1, W_2^1, \ldots, W_{31}^1\}$ (resp. $\{W_1^2, W_2^2, \ldots, W_{31}^2\}$) as the value set of the reduced attribute $\bar{w}^1$ (resp. $\bar{w}^2$), one may define the new attribute $\bar{w}$ as the "Multiplication" of $\bar{w}^1$ and $\bar{w}^2$: $\bar{w} = \bar{w}^1 \times \bar{w}^2$.

This new categorical attribute $\bar{w}$ has 961 values that we can denote by $\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_{961}$, where each value codes one subset $W_i^1 \times W_j^2$, $1 \leq i \leq 31$, $1 \leq j \leq 31$ (see above). Clearly, the obtained reduction level is not sufficient. Indeed, this attribute gives rise to $2^{960} - 1$ binary attributes. Therefore, a new application of $AVL_{0.5}$ is considered on the set of the rows of the contingency table $(\bar{w}, c)$, crossing this last attribute $\bar{w}$ with the attribute to be predicted c (see Fig.3).

The hierarchical classification methodology AVL enables to find "significant" partitions, of which the number of classes is of a given order. The last application of $AVL_{0.5}$ leads us to a significant partition of $\{\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_{961}\}$ into 12 clusters. This partition determines the final reduction of the value set of the initial attribute w. The associated qualitative (categorical) attribute that we may denote by $\bar{W}$ has 12 values. Each of them, which represents a class of the latter partition, corresponds to an union of sets, having the following form :

$\cup W_i^1 \times W_j^2$.

Let us denote by $E = \{e_1, e_2, \ldots, e_{12}\}$ the set of these 12 macro-values. The final part of the classification tree obtained by $AVL_{0.5}$, gives by its restriction on E, a classification tree on E. As an example which fixes ideas, consider Figure 4.

By itself the categorical attribute $\bar{W}$ provides $2^{11} - 1 = 2047$ binary attributes, corresponding to all the binary partitions of E. But, most of them are somewhat arbitrary

with respect to the prediction purpose. Each cluster of a such binary partition might be constitued by macro-values which are more or less dissimilar according to their statistical behaviour on the classes to predict. By considering the ultrametric organization fo this dissimilarity according to the above classification tree (see Fig.4) a partition as

$(\{e_3, e_9, e_{12}\}, \{e_1, e_2, e_4, e_5, e_6, e_7, e_8, e_{10}, e_{11}\})$

is, intuitively speaking, much less predictive than a partition as

$(\{e_4, e_5, e_7\}, \{e_1, e_2, e_3, e_6, e_8, e_9, e_{10}, e_{11}, e_{12}\})$.

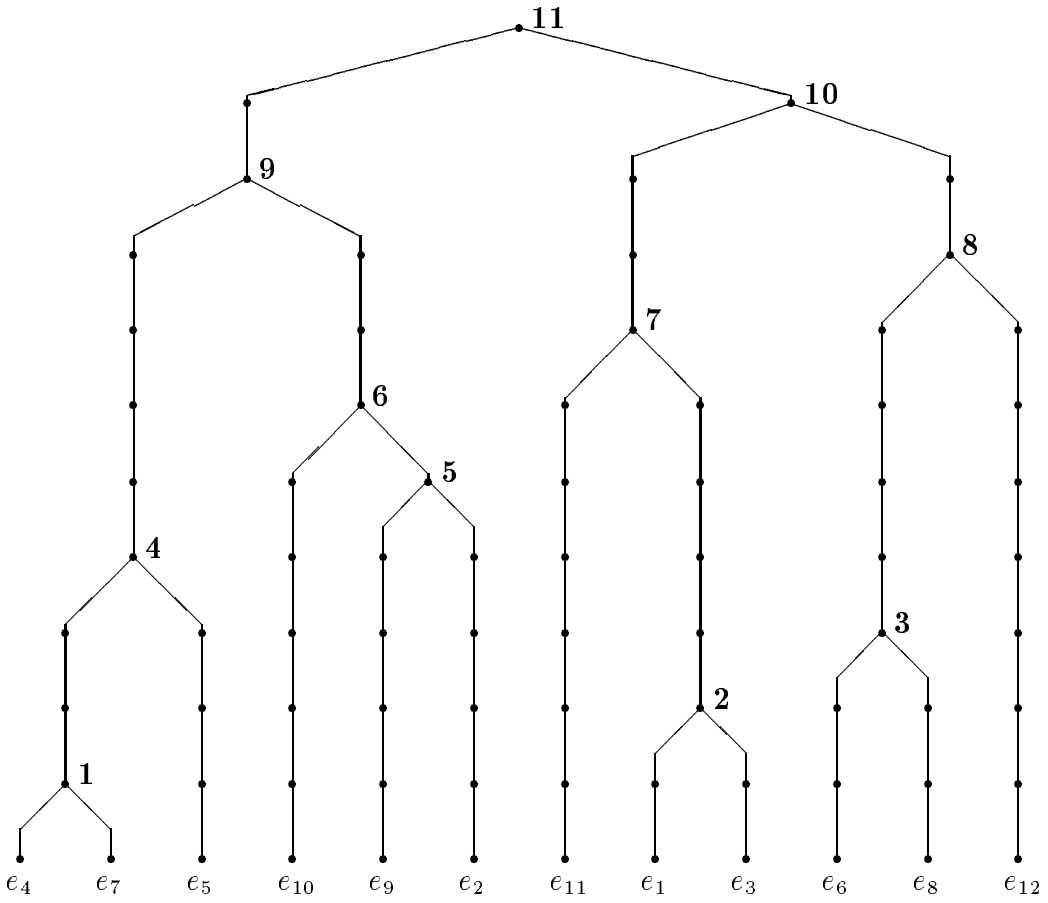

FIG. 4 - Example of a classification tree.

To avoid the unpredictive attributes, we only consider the partitions into two clusters, of which one of them is associated with one leaf or with one node of the hierarchical classification tree. Then, instead of using all of the $2^{12-1} - 1 = 2047$ binary attributes, we consider only those which correspond to the leaves and nodes of the hierarchy (see Fig.4 and Fig.5). Note that is was experimentally established in our application that the improvement of the prediction results would have been very negligible if we had taken all the 2047 binary attributes instead of those (about 20) recommended by the classification tree. These are the final attributes that we use in the decision tree program. The total reduction in complexity is therefore enormous :

$$(2^{20^4-1} - 1) \rightarrow (2^{961-1} - 1) \rightarrow (2^{12-1} - 1) \rightarrow \leq 22$$

Thus, the binary attributes that ARCADE chooses for the case above, are $a_1, a_2, \ldots, a_{21}$:

$$
\begin{array}{lll}
a_1: & (a_1 = 1 \text{ in } \{e_1\}) \text{ and } (a_1 = 0 \text{ in } \{e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}\}) & \text{(leaf)} \\
a_2: & (a_2 = 1 \text{ in } \{e_2\}) \text{ and } (a_2 = 0 \text{ in } \{e_1, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}\}) & \text{(leaf)} \\
\vdots & \vdots & \vdots \\
a_{12}: & (a_{12} = 1 \text{ in } \{e_{12}\}) \text{ and } (a_{12} = 0 \text{ in } \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}) & \text{(leaf)} \\
a_{13}: & (a_{13} = 1 \text{ in } \{e_4, e_7\}) \text{ and } (a_{13} = 0 \text{ in } \{e_1, e_2, e_3, e_5, e_6, e_8, e_9, e_{10}, e_{11}, e_{12}\}) & \text{(node 1)} \\
a_{14}: & (a_{14} = 1 \text{ in } \{e_1, e_3\}) \text{ and } (a_{14} = 0 \text{ in } \{e_2, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}\}) & \text{(node 2)} \\
\vdots & \vdots & \vdots \\
a_{20}: & (a_{20} = 1 \text{ in } \{e_6, e_8, e_12\}) \text{ and } (a_{20} = 0 \text{ in } \{e_1, e_2, e_3, e_4, e_5, e_7, e_9, e_{10}, e_{11}\}) & \text{(node 8)} \\
a_{21}: & (a_{21} = 1 \text{ in } \{e_4, e_7, e_5, e_{10}, e_9, e_2\}) \text{ and } (a_{21} = 0 \text{ in } \{e_{11}, e_1, e_3, e_6, e_8, e_{12}\}) & \text{(node 9)}
\end{array}
$$

FIG. 5 - Binary attributes used by ARCADE

This part of our method is similar to the Bit-Per-Category Encoding scheme (see [Almuallim & al., 1995] for a description of this and other methods for dealing with tree-structured attributes). In our method, however and by difference, the main contribution is in the construction of a tree-structured attribute, from a nominal one ; whereas in the work of [Almuallim & al., 1995], and other methods, the initial predictive attributes are already tree-structured.

2.3 The obtained results for the prediction of protein secondary structure

As mentionned in the abstract, ARCADE method consists in a powerful hybridation of the celebrated CART method [Breiman & al., 1984], by the above binarization technique of categorical attributes having a very large number of values (modalities). As anounced above, general expression of this method will be given in the next section. Thus the reader is assumed to be familiar with the CART method and specially with its pruning procedure depending on a quantitative parameter α, with respect to which the impurity of a node is compared with the impurity of its descendants.

Our database is constitued by a set $\mathcal{S} = \{s_i / 1 \leq i \leq 151\}$ of 151 non-homologous globular proteins ([Colloc'h & al., 1993]). The percentages of the three classes X, H and E are respectively, 46,6%, 29% and 24,4%. We used the jacknife procedure to assess the accuracy

of our method. Each one of the 151 proteins, $s_i / 1 \leq i \leq 151$, was successively chosen to be the test protein. The training set for this protein was constitued by the remaining 150 proteins ($\mathcal{O}_i = \mathcal{S} - \{s_i\}$).

This set $\mathcal{O}_i$ was then separated into 10 subsets, $\mathcal{O}_i^1$, $\mathcal{O}_i^2$, ..., $\mathcal{O}_i^{10}$, having each 15 proteins. Then, for $j = 1, 2, \ldots, 10$, the complementary set of $\mathcal{O}_i^j$ in $\mathcal{O}_i, \mathcal{O}_{i,j}$ ($= \mathcal{O}_i - \mathcal{O}_i^j$) is used to construct a decision tree $\mathcal{T}_{max,i,j}$, where each leaf is pure or contains no more than 5 observations. We have used various coefficients to choose the binary attribute which splits node of the decision tree (cf. [Lerman & Costa, 1996], [Costa, 1996]. The results that we present here are for the Gini coefficient. The predictive attributes used have been defined above and are deduced from 4-letter word attributes in a window of size 11. By associating with each 4-letter word attribute 2 binary attributes, we have in all $8 \times 22 = 176$ binary attributes. The pruning strategy was the same as in CART. From the sequence of subtrees obtained by pruning, the one that maximizes the prediction accuracy for the set $\mathcal{O}_i^j$, which has not yet been used, has been chosen. The corresponding pruning parameter is $\alpha(i, j)$.

For each $i = 1, 2, \ldots, 151$, we calculate the average $\alpha(i) = (\alpha(i, 1) + \alpha(i, 2) + \ldots + \alpha(i, 10))/10$. Afterwards, a decision tree $\mathcal{T}_{max}(i)$ is constructed on the set of 150 proteins $\mathcal{O}_i$. After pruning, the tree corresponding to $\alpha(i)$ is selected. To estimate the accuracy of this tree, we use the sequence s_i, which has not yet been used. A final operation (cf. [Lerman & Costa, 1996], [Costa, 1996]), which consists of a predicton correction index and is specific to this application, is used to improve the quality of the final prediction.

In the end, we have 151 percentages. The global quality of our method ARCADE is estimated as the average of this 151 values. This average can be an ordinal arithmetic mean ($\mathcal{Q}_{chain}$), or a weighted mean, where the weights correspond to the length of each protein ($\mathcal{Q}_{total}$). The percentages of each of the three classes (H, E, and X) were also calculated. For instance, $\mathcal{Q}_H^{obs}$ is the percentage of those elements, which have been observed in the H class, that are correctly predicted; $\mathcal{Q}_H^{pred}$ is the percentage of those elements, which have been predicted in the H class, that are correctly predicted.

The final results are :

Q_{chain} (%)	Q_{total} (%)	Q_H^{obs} (%)	Q_H^{pred} (%)	Q_E^{obs} (%)	Q_E^{pred} (%)	Q_X^{obs} (%)	Q_X^{pred} (%)
65.6	65.1	64.6	62.4	60	55.8	68.1	72.6

The results that we have obtained cannot be compared with the ones obtained by other methods that use different data sets. Nevertheless, the performances that we have obtained are amongst the ones obtained by the best methods. On the other hand, as pointed out by [Rost & Sander, 1993], for some of these methods the actual performances may be lower

if tested on non-homologous data sets or by using the jacknife procedure. The highest performance (70.8%) was obtained for the method PHD [Rost & Sander, 1993], which is a combination of several neural networks. This method uses an additional information, which we do not use, given by the multiple sequence alignements of homologous proteins. If a new protein has no homologs, the performance of this method fells considerably.

3 The ARCADE method

3.1 Introduction

As clearly expressed above, ARCADE method is an hybridation of CART method [Breiman & al., 1984] by mainly a binarization technique of predictive categorical attributes having very large number of values (modalities). This hybridation includes also a large family of association cœfficients which enable to choose the binary attributes which split the nodes of the decision tree ([Lerman & Costa, 1996], [Costa, 1996]). As shown above the binarization procedure can be decomposed into four steps :

(i) Factorisation of the predictive categorical attribute v into two (as above) or more than two factors.

(ii) Classification of the set value of each factor on the basis of the contingency table, crossing this factor with the categorical attribute to predict c.

(iii) Multiplication of the reduced factors and setting up the contingency table which crosses the multiplied attribute with the attribute to predict.

(iv) Hierarchical classification of the set value of the new multiplied attribute, represented by the rows of the latter contingency table.

Thus the specification of ARCADE method can be figured by the diagram pictured in figure 6.

Notice that only the above step (iv) has to be considered in case where the number $\mathcal{L}(v)$ of values of the predictive categorical attribute v, is not too large. Because the contingency table crossing v with the attribute to predict becomes statistically consistent. Moreover the size of the row set of the contingency table becomes tractable by a clustering algorithm.

3.2 The Binarization method

The binarization procedure has been expressed (see section 2.2) in the context of our application. Formal and general description of this method will be given here. Different cases and subcases have to be distinguished according to the mathematical structure of the value

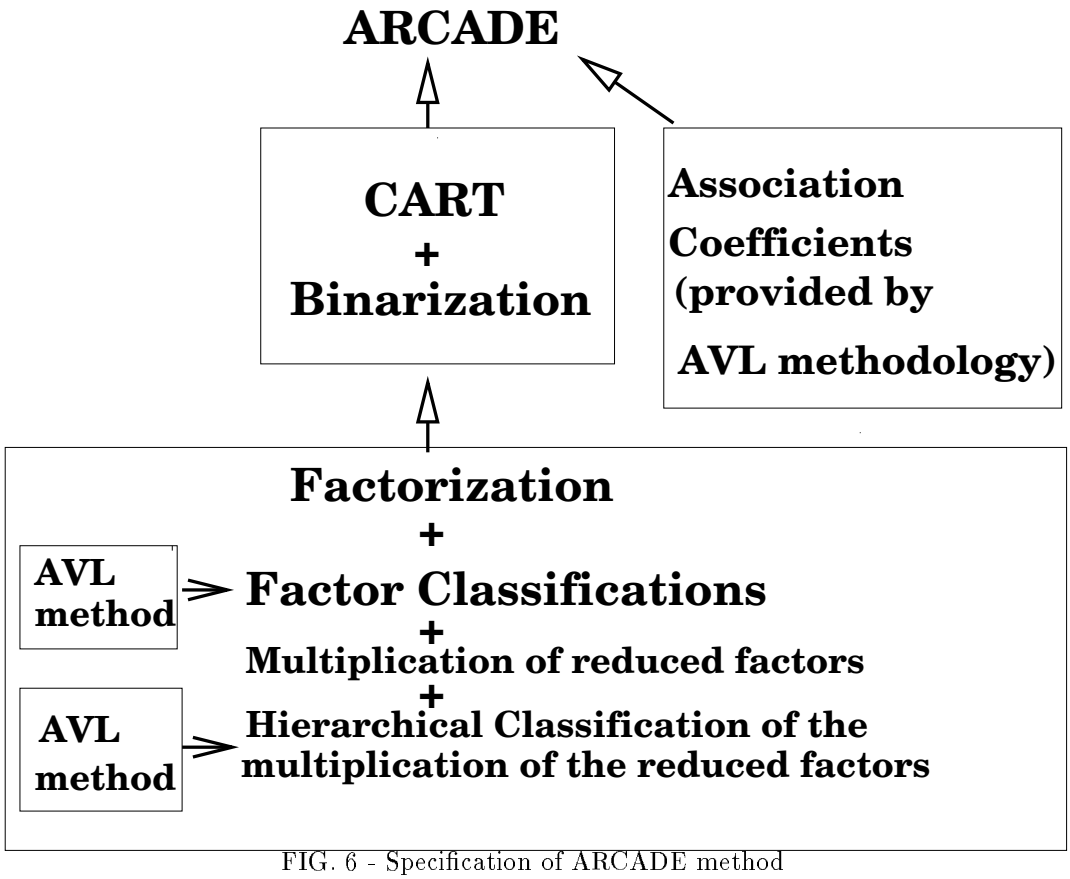

FIG. 6 - Specification of ARCADE method

set $\mathcal{E}(v)$ of the categorical predictive attribute v. The first case to be considered is that where $L = L(v) = card[\mathcal{E}(v)]$ is not too large, in order to make statistically consistent the contingency table crossing v with the attribute to predict c. In case where L is too large, two subcases will be distinguished. For the former the attribute v is multidimensional by nature and can be defined by a sequence $(v^1, v^2, \ldots, v^j, \ldots, v^q)$ of q attributes. As an example in our application, the 4-letter word attribute occupying the positions 5,6,7 and 8 of a window of size eleven is defined as a sequence (v^5, v^6, v^7, v^8) (see section 2.1 above). For this former subcase the value set $\mathcal{E}(v)$ can be decomposed as a cartesian product of sets. For the latter subcase $\mathcal{E}(v)$ has not any structure.

Let us designate by $\{c_1, c_2, \ldots, c_k, \ldots, c_K\}$ the value set of the categorical (qualitative nominal) attribute c to predict. c_k codes one class or concept to recognize from the description given by the predictive attributes, $1 \leq k \leq K$. In our application the value set of

c is $\{X, H, E\}$ and $K = 3$. If v is one of the categorical predictive variables, let us denote by $\{v_1, v_2, \ldots, v_L\}$ its value set, indicated above by $\mathcal{E}(v)$. The statistical basic information used for clustering $\mathcal{E}(v)$ is the contingency table crossing v and c (see Fig.7)

v \ c	c_1	c_2	$\ldots$	c_K
v_1				
v_2				
$\vdots$				
v_L				

FIG. 7 - Contingency table crossing v with c.

3.2.1 The case where L is not too large

In this case – where the above contingency table is consistent – the binarization procedure of ARCADE begins by grouping the set of the values of the predictive attribute v in order to reduce the number of binary splits to be considered. These values are grouped into clusters of similar values, so that each cluster will constitute a new value of a new nominal attribute $\bar{v}$. More precisely, the similarity between the values, is relative to the discrimination of the values of the concept attribute c. That is the reason for applying a classification method to the rows of the contingency table above.

Definition. Denote by $\{1, 2, \ldots, L\}$ the value set of a given attribute v, with L values ; and consider the set G^L of all partitions of $\{1, 2, \ldots, L\}$, into more than one cluster and for which at least one cluster includes more than one element. We define a *grouped attribute* $\bar{v}_g$ associated with g (belonging to G^L), as the qualitative attribute of which the values are defined by the g-clusters. More precisely, denoting by $j_k = \{i_1, i_2, \ldots, i_u\}$ a given g-cluster, j_k is the value of $\bar{v}_g$ on a given object, if and only if i_1 or i_2, ..., or i_u, is the value of v on this object.

Obviously, this new attribute $\bar{v}$ has fewer values than the initial attribute v ; indeed, we can choose, in a proper way, the number of values to use in $\bar{v}$ according to the complexity that we can accept to binarise it. These new synthetic values, which we shall call macro-values, must be created in accordance with the discrimination principle underlying the construction of a decision tree. To do this, we use an automatic classification of the rows of the contingency table crossing the attribute v with the attribute c to be predicted.

There are essentially two types of automatic classification ; hierarchical and non-hierarchical. In our case we have employed the hierarchical classification method AVL ("Analyse de la Vraisemblance des Liens"). This method has been introduced by [Lerman, 1970, 1981, 1987, 1992_a, 1992_b, 1993] and developped by him and his collaborators. It is implemented in the computer program CHAVL [Lerman, Peter & Leredde, 1993-1994]. We have used a specific

form for the agregation criterion, called $AVL_{0.5}$, and proposed by [Bacelar-Nicolau, 1985]. The relevancy of this methodology is motivated by the following reasons ;

(i) It is underlined by a very general constructive method of probabilistic similarity indices between combinatorial or logical structures, on a given object set $\mathcal{O}$. These structures are interpreted in terms of relations on $\mathcal{O}$, eventually weighted ; and then, the numerical case is comprised. Description of concepts observed on $\mathcal{O}$ is also taken into account in this approach, which is extremely general with respect to the data structure. On the other hand, it introduces the Likelihood concept into the Resemblance notion.

(ii) This method allows ([Lerman, 1983], [Lerman & Ghazzali, 1991]) the identification of the significant levels and nodes. A significant level does correspond to a stable partition ; and a significant node to the completion of a cluster.

(iii) The hierarchical tree (on the value set of a given attribute) determines an ultrametric distance between the macro-values.

The two latter points are very important to define, in a relevant and reduced manner, the binary attributes that are going to be used in the decision tree.

Thus the ARCADE method performs an $AVL_{0.5}$ hierarchical classification on the set $\mathcal{E}(v)$ of the L values of our predictive attribute v and then chooses a significant partition with J clusters. As an example, consider here our application where $L = 400$ for an attribute v defined by a 2-letter word description and assume $J = 12$. This allows to define new attribute $\bar{v}$ with J macro-values. By this manner the first stage of ARCADE reduces significantly the complexity of the problem by replacing a predictive attribute with L values by another one with J values (J much less than L). Each of the J values of this new attribute is a cluster of the initial set $\mathcal{E}(v)$ of L values.

Additional and very important complexity reduction is provided by the ultrametric structure of the restriction of the classification tree on the J macro-values set. The meaning of this reduction is given at the end of section 2.2 which can be retaken here, by considering to fix ideas $J = 12$ and then by denoting $\{e_1, e_2, \ldots, e_{12}\}$ the set of the J macro-values. Notice that in case where the latter classification tree is binary, there are in all $(2J - 1)$ binary attributes which are used in the decision tree construction.

3.2.2 The case where L is too large

Now we consider the case where the number of values of the predictive attribute v is too large for the above contingency table to be consistent. In other words, the size of the learning set $\mathcal{O}_l$, is not large enough to assure the statistical consistency needed for the calculations, because the contingency table (see Fig.7) is too hollow. As expressed above, we are going to distinguish two subcases according to the structure of v ; multidimensional or not.

3.2.2.1 The attribute v is multidimensional

We assume here that the predictive attribute v is defined by a sequence of q attributes $(v^1, v^2, \ldots, v^p, \ldots, v^q)$. Let us designate by $A_p = \mathcal{E}(v^p)$ the value set of the p^{th} attribute v^p, $1 \leq p \leq q$. In these conditions, the value set of v is given by the cartesian product

$$\mathcal{E}(v) = A_1 \times A_2 \times \ldots \times A_p \times \ldots \times A_q$$

In our application where the attributes are defined by q-letter words taking position in a window of a given size ($> q$), all the sets A_1, A_2, ..., A_{q-1} and A_q are identical to a set A of 20 letters representing the 20 amino acids and then $\mathcal{E}(v) = A^q$.

Consider the set $Q = \{1, 2, \ldots, p, \ldots, q\}$ of the above subscripts and let

$$\pi(Q) = \{Q_i / 1 \leq i \leq r\}$$

be a partition of Q into r non empty classes. One may denote Q_i as following :

$$Q_i = \{p_{i1}, p_{i2}, \ldots, p_{ic_i}\},$$

$1 \leq i \leq r$, where

$$c_1 + c_2 + \ldots + c_r = q$$

We define the $\pi(Q)$-FACTORIZATION of order r of v as the grouping of the components of v into r subsequences according to the partition $\pi(Q)$. More precisely the attribute v is viewed as the following sequence of r attributes

$$(u^1, u^2, \ldots, u^i, \ldots, u^r)$$

where

$$u^i = (v^{p_{i1}}, v^{p_{i2}}, \ldots, v^{p_{ic_i}}),$$

$1 \leq i \leq r$.

Now the value set of u^i is defined by

$$A_{p_{i1}} \times A_{p_{i2}} \times \ldots \times A_{p_{ic_i}},$$

$1 \leq i \leq r$.

The cardinality of the latter is given by

$$c(p_{i1}, \ldots, pic_i) = \prod_{1 \leq j \leq c_i} card(A_{p_{ij}})$$

where $card(A_{pij})$ is the cardinality of A_{pij}.

The above factorization is all the better balanced as the integer numbers $c(p_{i1}, \ldots, pic_i)$ are close. A measure of the balance of the factorization can be

$$\sum_{\{(i,i')/1 \leq i < i' \leq r\}} | \, c(p_{i1}, \ldots, p_{ic_i}) - c(p_{i'1}, \ldots, p_{i'c'_i}) |$$

In the context of our application reconsider the example of the 4-letter word attribute occupying the positions 5,6,7 and 8 of a window of size eleven. The initial attribute w is defined by (v^5, v^6, v^7, v^8). Q is equal to $\{5, 6, 7, 8\}$ and the partition $\pi(Q)$ is into two classes comprising two consecutive elements each :

$\pi(Q) = \{\{5, 6\}, \{7, 8\}\}$

Hence, $w = (u^1, u^2)$ where $u^1 = (v^5, v^6)$ and $u^2 = (v^7, v^8)$. In these conditions $c(5, 6) = c(7, 8) = 20 \times 20 = 400$.

The ordered partition of the sequence $(v^1, v^2, \ldots, v^p, \ldots, v^q)$ into the sequence $(u^1, u^2, \ldots, u^r)$ of subsequences (see above) can be guided by formal considerations. Thus as in our application, u^i can be associated with a sequence of consecutive attributes v^p. For this case u^i is defined as follows :

$u^i = (v^{c_{i-1}+1}, v^{c_{i-1}+2}, \ldots, v^{c_{i-1}+c_i})$

If there is no formal constraints for determining the subsequences u^i from the sequence v of the q attributes v^p, the most interesting consists of determining a clustering of $\{v^1, v^2, \ldots, v^p, \ldots, v^q\}$ according to the observed similarities between the predictive descriptive attributes v^p, $1 \leq p \leq q$. The hierarchical classification methodology AVL enables to provide such a clustering by associating with each attribute v^p, $1 \leq p \leq q$, a partition on the described training set. More clearly, the partition induced by v^p is specified according to the taken value in $A_p = \mathcal{E}(v^p)$. And then, clustering by similarity the attribute set $\{v^1, v^2, \ldots, v^p, \ldots, v^q\}$ consists of clustering by similarity a set of q partitions on the training set.

We associate now with each attribute u^i, $1 \leq i \leq r$, crossing its value set with the value set $\{c_1, c_2, \ldots, c_k, \ldots, c_K\}$ of the attribute c to predict. This contingency table has $c(p_{i1}, \ldots, p_{ic_i})$ rows and K columns (see above). On the basis of the latter, we apply the hierarchical classification $AVL_{0.5}$ on the row set. The purpose consists in obtaining from each classification tree resulting from each contingency table a significant partition of which

the number of classes is around a given size s. s is chosen a priori such as a contingency table distributing the set of observations on $s^r \times K$ entries has the necessary accuracy and consistency.

Consider the partition which clusters the values of u^i. A class of this, is a subset of $A_{p_{i1}} \times A_{p_{i2}} \times \ldots \times A_{p_{ic_i}}$ (see above). Each such subset determines one single value of the reduced grouped attribute $\bar{u}^i$ deduced from u^i (see **Definition**).

Let us designate by

$$\pi(u^i) = \{B_1^i, B_2^i, \ldots, B_j^i, \ldots, B_{k_i}^i\}$$

the partition of the value set of u^i, where the number of classes k_i is close to s, $1 \leq i \leq r$. One may denote by $\{b_1^i, b_2^i, \ldots, b_j^i, \ldots, b_{k_i}^i\}$ the value set of $\bar{u}^i$, where b_j^i represents the set B_j^i, $1 \leq j \leq k_i$.

Now, we consider a MULTIPLICATION in a given order of the set $\{\bar{u}^1, \bar{u}^2, \ldots, \bar{u}^i, \ldots, \bar{u}^r\}$ of the reduced attributes. This operation is in a some manner inverse of the FACTORIZATION one. More explicitely, we consider here the attribute that we denote $\bar{u}$ and defined by the ordered sequence $(\bar{u}^1, \bar{u}^2, \ldots, \bar{u}^i, \ldots, \bar{u}^r)$. A given value of $\bar{u}$ is defined by a sequence of values b_j^i having the following form :

$$(b_{j_1}^1, b_{j_2}^2, \ldots, b_{j_i}^i, \ldots, b_{j_r}^r),$$

$1 \leq j_i \leq k_i$, $1 \leq i \leq r$. This value represents the cartesian product

$$B_{j_1}^1 \times B_{j_2}^2 \times \ldots \times B_{j_i}^i \times \ldots \times B_{j_r}^r$$

Hence the reduction obtained at this level for the initial attribute $v = (v^1, v^2, \ldots, v^q)$ consists of identifying all its values belonging to the preceding set.

For the following reduction stage, the contingency table crossing the value set of the above multiplied attribute $\bar{u}$ with the value set of the attribute c to predict. This table has $k_1 \times k_2 \times \ldots \times k_i \times \ldots \times k_r$ rows and K columns. Since each k_i is near s, the number of rows is near s^r.

Finally, as in our application where r was equal 2 (see section 2.2), hierarchical classification $AVL_{0.5}$ is performed on the row set of the last contingency table. By retaining a significant partition [Lerman & Ghazzali, 1993] we define a new synthetized categorical attribute $\bar{U}$ of which the respective values are associated with the classes of this partition. Note that a given class has the following form

$$\cup B^1_{j_1} \times B^2_{j_2} \times \ldots \times B^i_{j_i} \times \ldots \times B^r_{j_r}$$

Note also that the value set of $\bar{U}$ has been illustrated in our application by a set of 12 values, designated by $\{e_1, e_2, \ldots, e_{12}\}$ (see section 2.2).

As described in section 2.2 the ultimate macro-values that we keep correspond to the leaves or nodes of the restriction of the classification tree to the classes of the retained previous partition. This restriction corresponds clearly to the latter part of the classification tree. To each such macro-value a binary attribute is associated. And consequently, if J is the number of classes of the latter partition, there will be at most $(2J-1)$ binary attributes associated with the initial attribute v which will intervene in the tree decision building.

We assume here that the value set $\mathcal{E}(v)$ of v has not any specific structure. We code by the first L positive integers $\{1, 2, \ldots, l, \ldots, L\}$ this value set. We are now going to describe how to apply the FACTORIZATION procedure to $\mathcal{E}(v)$. Let us start by a very small and simple example where $L = 4$ and $r = 2$ (factorization of order 2). The sequence of values $(1, 2, 3, 4)$ is coded by a sequence of bidimensional vectors (i, j) lexicographically ordered where each component is included in the interval $[1, 2]$. The first and second components are respectively the values of two created attributes v^1 and v^2. More explicitely the correspondence between the values is :

v	v^1	v^2
1	1	1
2	1	2
3	2	1
4	2	2

A factorization of order 2 is generally sufficient for the current applications. We only consider here this case. A generalization for any r is easy to concieve. Two cases have to be distinguished according to the fact that L is a perfect square or not. For the former case L can be written L'^2, where L' is a positive integer. Then and clearly the above correspondance table between the values of v and those of the factors v^1 and v^2, becomes :

v	v^1	v^2
1	1	1
2	1	2
⋮	⋮	⋮
L'	1	L'
L'+1	2	1
⋮	⋮	⋮
2L'	2	L'
⋮	⋮	⋮
L	L'	L'

By this technique the value set $\mathcal{E}(v)$ is decomposed into a cartesian product $\mathcal{E}(v^1) \times \mathcal{E}(v^2)$ where $\mathcal{E}(v^1)$ and $\mathcal{E}(v^2)$ are respectively the value sets of created attributes v^1 and v^2. These factorize the initial attribute v. Nevertheless this decomposition is arbitrary and depends on the coding of the value set of v. Since it is intuitively desirable to make v^1 and v^2 as independent as possible, we suggest to assign randomly the codes $1, 2, \ldots, L$ to the values of v.

Let us now consider briefly the case where the number L of the values of the attribute v, is not a perfect square. The factorization procedure of order 2 can still be applied. For this purpose define $L' = [\sqrt{L}] + 1$, where $\sqrt{L}$ is a real positive number and where $[x]$ designates the integer part of the supposed positive number x. We have $L'^2 > L$. Consider now the following set of bidimensional vectors

$$\{(i,j)/1 \leq i \leq L',\ 1 \leq j \leq L'\}$$

that we suppose totally lexicographically ordered as in the above table. And now make one increasing correspondence between the sequence $(1, 2, \ldots, l, \ldots, L)$ and that defined by the ordered pairs. Since $L'^2 > L$, the latest part of the sequence of ordered pairs do not correspond to any of the original values of v. This shall not bring any serious problem in the treatment of the factors v^1 and v^2, where the values of v^1 and v^2 are respectively defined by the first and the second components of the ordered pairs.

Now, according to the scheme described above, we associate with v^1 and v^2 a pair of contingency tables crossing respectively v^1 and v^2 with the attribute to predict c. The common size of these tables is $L' \times K$ (L' rows and K columns). $AVL_{0.5}$ is then applied to the row set of each of the contingency tables. From each of both classification trees we retain a significant partition having approximatively s classes. As indicated above s is empirically chosen such as and a priori a contingency table with $s^2 \times K$ entries, is enough accurate and statistically consistent. The retained partitions induce the reduction of v^1 and v^2 into synthetized attributes $\bar{v}^1$ and $\bar{v}^2$. By denoting $\{B_1^1, \ldots, B_j^1, \ldots, B_{k(1)}^1\}$ and $\{B_1^2, \ldots, B_{j'}^2, \ldots, B_{k(2)}^2\}$ these partitions, B_j^1 (resp. $B_{j'}^2$) corresponds to one single value of

$\bar{v}^1$ (resp. $\bar{v}^2$), $1 \leq j \leq k(1)$ (resp. $1 \leq j' \leq k(2)$).

As before the next step consists of defining a new attribute $\bar{u}$ which results from the multiplication of $\bar{v}^1$ and $\bar{v}^2$: $\bar{u} = (\bar{v}^1, \bar{v}^2)$. A given value of $\bar{u}$ can be written as the cartesian product $B_j^1 \times B_{j'}^2$, $(1 \leq j \leq k(1),\ 1 \leq j' \leq k(2))$. Note here that some few values of (v^1, v^2) falling in some of the sets $B_j^1 \times B_{j'}^2$ $(1 \leq j \leq k(1),\ 1 \leq j' \leq k(2))$ might not correspond to any original value of the attribute v. This occurs when $L'^2 > L$, for coding reasons. These non representative values of (v^1, v^2) are simply ignored.

The following steps of the reduction process are continued as previously :

(i) setting up the contingency table ($k(1) \times k(2)$ rows and K columns) which crosses the attributes $\bar{u} = (\bar{v}^1, \bar{v}^2)$ and c ;

(ii) performing hierarchical classification $AVL_{0.5}$ on the row set of the latter contingency table ;

(iii) gathering a significant partition of which the number of classes is of a given order and considering the restriction of the classification tree on the set of the classes of this partition (this restriction corresponds to the end part of the classification tree) ;

(iv) associating with each leave or node of the preceding truncated classification tree a macro-value of a new attribute $\bar{U}$, each such macro-value gives rise to a binary attribute which will intervene in the binary decision tree construction.

3.2.3 Comparison between ARCADE and CART methods

We shall now give some elements for the comparison between ARCADE and CART in terms of their respective complexities. In this comparison we will refer to our application (see section 2). Let us start by considering the most favourable case for CART in which the number K of classes to predict is equal to 2 and where the used criterion to split a node in the tree decision construction has an inertial nature, as the Gini cœfficient ([Lerman & Costa, 1996], [Costa, 1996]). Under these conditions, in this method, with a categorical predictive attributes v having L values, one associates $L-1$ binary attributes. These are obtained by successive cuttings according to a sorting of the value set of v. This sorting is established at each node of the decision tree, according to the relative frequency of the class 1 (or 2) to be predicted on each value of v. These relative frequencies are empirically determined from the subset of the learning set which underlines the concerned node. The order of the number of nodes of the decision tree is $O(n)$ (Landau notation), where n is the size of the learning set. on the other hand the computational complexity for establishing the mentionned sorting on L values, is in $O(L\ log_2\ L)$.

Therefore the computational complexity for setting up the binary attributes we have to evaluate, is in $O(nL\ log_2\ L)$. This evaluation is based upon an association cœfficient between categorical attributes as the Gini cœfficient ([Lerman & Costa, 1996], [Costa, 1996]). Another type of computation complexity is defined by the number of binary attributes that we have to check at each node. This number is here $L-1$. In these conditions it is obvious that it is absurd, for statistical and complexity reasons, to apply the method provided by CART for binarization our initial predictive attributes with 20^4 values.

Our data is constituted by 151 non homologous proteins ; $S = \{s_i / 1 \leq i \leq 151\}$. To each s_i we associate the complementary set $\mathcal{O}_i = S - \{s_i\}$. The idea now is to learn with these sets of 150 proteins and to estimate the true performance of the final decision tree with s_i ; this is done 151 times. The binarization ARCADE method is applied once on the whole set of 151 proteins, instead of applying it 151 times by excepting at each time the sequence s_i on which the ith decision tree is evaluated, $1 \leq i \leq 151$. Indeed we have shown [Costa, 1996] that the difference in the final permormances – between one application of our binarization method or applying it more rigorously 151 times – is smaller than $0,5\%$.

In CART method the number of binary attributes issued from a categorical predictive attribute v with L values is equal to $L-1$ and then is linear whith respect to L. In ARCADE method the reduction with respect to L is very important. It is about $1,5 \times 10^{-4}$ in our application where a factorization of order 2 has been considered. Moreover the binary attributes constitued by ARCADE method have a statistical meaning.

Consider a factorization of order r equal to 2 and let us evaluate in this case the computational complexity of the ARCADE binarization. The automatic classification method AVL is of complexity $O(l^2\ log_2(l))$ for a set of size l. This method has to be employed three times before the construction of the decision tree. Thus in our application the hierarchical classification has been applied twice in order to cluster the set of the rows of two contingency tables having he same size : 400 rows $\times$ 3 columns. Clearly, these two classifications can be performed in parallel. The third classification concerns the set of rows of a contingency table comprising 961 rows. This size is 2.4 times the common preceding size of the respective row sets of the previous contingency tables. More generally, we have to apply in parallel twice the hierarchical classification included in AVL, on respectively the row sets of two contingency tables having the same size $\sqrt{L}$. The computational complexity for this treatment is $O(L\ log_2(\sqrt{L}))$. The third and last application of the classification algorithm concerns a row set of which the size is of the same order as $\sqrt{L}$. For an illustration suppose as in our application, this size to be less than $4\sqrt{L}$, then the complexity is bounded by $O(32L\ log_2(\sqrt{L}))$. This complexity is lower than that $O(nL\ log_2(L))$ reached in the CART method in order to determine the binary attributes associated with a categorical attribute v having L values.

Now we could want to employ our idea of factorization in CART. For this purpose and for $r = 2$, we could start by factorizing a given predictive attribute v with L values ($e.g. \sqrt{L} = 400$). Thus, in each node of the decision tree, there are two sets of $\sqrt{L}$ binary attributes to evaluate. The best binary attribute on each set is chosen and then these two binary attributes are multiplied giving rise to 4 binary attributes. The best among these is finally taken for splitting the concerned node. Therefore the complexity on each node, in terms of the number attributes to consider, is $2\sqrt{L}+4$ (e.g. $2 \times 400 + 4 = 804$). Nevertheless there is a statistical consistency problem when we go down in the decision tree.

More deeply one may try to use as in our approach in addition of the factorization technique the classification idea but in being the nearest CART method techniques. In this method, the justification of the binary attributes determined from a categorical one is given by [Fisher, 1958].

By using dynamic programming, the Fisher algorithm provides the best partition of the value set of the categorical attribute, into a given number of classes and according to an inertial criterion. In CART two classes are requested. By demanding after factorization s classes for each factor, we may reproduce our reduction process (see section 3.2.2), but by using Fisher algorithm instead of AVL method. But in this manner, if J designates the final number of classes obtained in the last application of the classification algorithm, we cannot reduce from $(2^{J-1} - 1)$ to $(2J - 1)$, the number of binary attributes to consider in the decision tree construction.

Moreover as mentionned above, it is of importance to notice that the use of the Fisher classification algorithm is constrained by two conditions. The former consists of the inertial nature of the criterion employed for splitting the nodes of the decision tree. The latter which is the most serious, fixes $K = 2$ classes to predict. The solution proposed in CART for K increasing is exponential with respect to K, whereas the complexity of ARCADE is independent of K.

4 The AVL hierarchical classification method

4.1 Introduction

The most distinctive characteristics of this approach have been expressed above (see section 3.2.1). Only a brief general description can be given here. For more substantive introductory texts in english language see [Lerman, 1991] and [Lerman, 1993].

AVL (Analyse de la Vraisemblance des Liens) is a methodology for grouping data into "significant" classes and subclasses, using an algorithm of hierarchical classification. The latter is built in an ascendant way from the leaves to the root by successive agglomerations. However, this approach ha many more than only algorithmic aspects. Its elaboration is at the

intersection of three fields : "combinatorics", "logic" and "non-parametric statistics". In fact, it gives a very general view of the "data" and of their automatic synthesis. Additionally and mainly, this method introduces a most original notion of "statistics" for measuring statistical relationships and proximities, namely, the "likelihood" concept. Thus, we set up the "likelihood" notion as part of the "resemblance" notion. This principle also underlies the "information theory" formalism, in which the higher the amount of information quantity, the more unlikely is the event concerned. In our case the events correspond to the observed relations.

4.2 General concepts and probabilistic association

The AVL hierarchical classification method concerns any mathematical or logical type of a data table T giving description from empirical observation or knowledge. We begin by distinguishing two main types for T. For the former where the description concerns a set O of elementary objects, T crosses O with a set A of descriptive attributes (we also say "variables" or "parameters"). For the latter the description concerns a set C of categories (we also say concepts or classes) and the data table T crosses C with a set A of descriptive attributes.

The first important idea consists of interpreting a descriptive attribute of an object set O in terms of a q-ary relation on O that we represent by a structured subset of O^q. This relation can be weighted as for a numerical attribute on O or a similarity attribute on O. In practice, it is sufficient to consider q=1 or 2 or 4, in the qualitative or quantitative data analysis that one may have to deal with.

In these conditions, associate with each descriptive attribute a^j represented by the jth column of the data table T, a relation R_j of which the arity q_j is determined by the structure of the value set $\mathcal{E}(a^j)$, $1 \leq j \leq m$. Assume here for simplicity reasons in the presentation that all the q_j are equals. Indeed, this occurs the most frequently. As an example, consider the case where all the variables a^j $(1 \leq j \leq m)$ are categorical (nominal qualitative) without any particular structure behind the value set $\mathcal{E}(a^j)$, $1 \leq j \leq m$. For this case, each R_j induces a partition relation on O. Finally the data table T of which the rows (respectively the columns) are labelled by the elements of the object set O (respectively the attribute set A), can be expressed as a relational system of Tarski [Tarski, 1954] :

$T =< O; R_1, R_2, \ldots, R_j, \ldots, R_m >$

The AVL method is then able to handle the two dual problems :

(i) classification by proximity of the relation set $\{R_j / 1 \leq j \leq m\}$ on O

(ii) classification by proximity of the object set O described by the set $\{R_j/1 \leq j \leq m\}$ of relations that we consider a priori to have the same importance.

A final stage of this method studies by means of association cœfficients the relative positions of the above two classifications [Lerman, 1981, 1992a, 1992b].

Description of a set C of concepts or categories by a set A of descriptive attributes leads us to introduce another type of system that we denote by

$S =< C; R_1, R_2, \ldots, R_j, \ldots, R_m >$

Herein the data correspond to the statistical distribution of each R_j representing $a^j,\ 1 \leq j \leq m$, on each concept (category) c belonging to C.

To be more explicit consider a given entry (c_i, a^j) of the data table T labelled by the cartesian product $C \times A,\ 1 \leq i \leq card\ C,\ 1 \leq j \leq m$. Then you have to imagine this entry containing the empirical statistical distribution of a^j on c_i ; and consequently, many cells might be necessary to describe this distribution, $1 \leq i \leq n,\ 1 \leq j \leq m$.

Therefore we may note that the data structure concerned by a contingency table or, more generally, by an horizontal juxtaposition of contingency tables, is a particular case of the system S. More precisely, a single contingency table can be formalized by a system S comprising only one partition relation $R(m = 1)$ associated with a nominal qualitative attribute a : $S =< C, R >$. As defined, the above data table T includes only one conceptual column. But each of its entries is supposed to contain the empirical distribution of the attribute a on one of the categories (concepts). And then, if $K(a)$ denotes the number of values of the attribute a, we associate with $S =< C, R >$ a contingency table TC comprising $N = card(C)$ rows and $K(a)$ columns. TC contains concretely the mentionned empirical distribution. Clearly the $(i,k)th$ entry of this table contains the number of times where the kth value of a occurs in the category c_i, $1 \leq i \leq N$, $1 \leq k \leq K(a)$. As a matter of fact, the contingency table TC gives the class cardinalities of the partition resulting from the crossing of two partitions on the observation set. The former is induced by a categorical attribute that we denote by W of which the value set is C and the latter by the attribute a. In our application problem W is defined by a word (of two or four letters) and the attribute a is associated with the protein secondary structure observed at a given position. Thus $K(a) = 3$.

Also for the data structure defined by the system S, the AVL method assumes the two dual problems :

(i) classification by proximity of the set of the relations $\{R_j/1 \leq j \leq m\}$ observed from their respective statistical distributions on the different concepts or categories c of C ;

(ii) classification by proximity of the concept (category) set C described by the statistical distributions of each of the relations $R_1, R_2, \ldots, R_m$, considered a priori with the same importance.

Finally, the relative position of these two hierarchical classifications is studied by means of the association cœfficients concieved and developed in the AVL context [Lerman 1981, 1992a, 1992b].

Note that in our application case where S is reduced to a very simple version, we are only interested in the classification of the set C.

Let E designate the set to be classified. E can be a set O of elementary objects (respectively a set C of categories or concepts) or, dually a set A of descriptive attributes. This description is assumed formalized by a system T (respectively a system S) (see above).

The elaboration of an association cœfficient (one can also say similarity or resemblance index) depends first on the common arity of the relations R_j, $1 \leq j \leq m$, representing the descriptive attributes. And also it depends on the specific combinatoric type of the relations R_j, $1 \leq j \leq m$. The latter is induced by the common nature of the structure of the value scales of the attributes a^j, $1 \leq j \leq m$. For example, if the descriptive attributes are numerical, then the R_j, $1 \leq j \leq m$, are weighted unary relations on the object set O. But if the attributes a^j, $1 \leq j \leq m$, are nominal qualitative, then the relations R_j, $1 \leq j \leq m$, are binary and induce each a partition on O. Moreover, this latter case is distinct from that where the attributes R_j, $1 \leq j \leq m$, are ordinal qualitative. For this, each of the binary relation R_j, $1 \leq j \leq m$, induces a total (or partial) preorder on O.

There are many others cases which occur in concrete data analysis. However, we have set up in the AVL context general principles and a general constructive method of association coefficients between descriptive attributes (respectively similarity indices between objects or categories). In this constructive method two principal and dual cases are distinguished. The former concerns the pairwise comparison between attributes and the latter, between objects or categories [Lerman 1987, 1992a, 1992b]. We only can give here very general idea of this method. It can be decomposed into the following steps, relative to the set E to be classified :

(i) from combinatorial or geometrical considerations, establish a raw similarity index q for the comparison two by two of the elements of E : $\{q(x,y)/\{x,y\} \in P_2(E)\}$, where $P_2(E)$ is the set of unordered element pairs of E ;

(ii) consider a probabilistic "hypothesis of no link ($h.n.l$)", more common called "hypothesis of independence" under which, we associate with the relation set $\{R_j/1 \leq j \leq m\}$, a set $\{R*_j/1 \leq j \leq m\}$ of independent random relations, having respectively the same structural

and cardinal characteristics as them ;

(iii) $q(x*, y*)$ being the random raw index associated with $q(x, y)$ under the $h.n.l$, consider a standardized version $Q(x, y) = (q(x, y) - \mathcal{E}(q(x*, y*)))/\sqrt{var(q(x*, y*))}$ of $q(x, y)$, where $\mathcal{E}$ and var designate respectively the mathematical expectation and the variance, $\{x, y\} \in P_2(E)$;

(iv) compute the probabilistic association coefficient (similarity) with the expression

$$P(x, y) = Prob\{Q(x*, y*) \leq Q(x, y)/h.n.l\}$$

More precisely, $\mathcal{E}(q(x*, y*))$ and $var(q(x*, y*))$ are mathematically computed in the different cases (see above mentionned references). On the other hand, except few structural situations, the limit distribution of $Q(x*, y*)$ is normal $N(0, 1)$. Then, $P(x, y)$ is computed – in most cases – by the following approximation :

$$P(x, y) \simeq \Phi(Q(x, y)),\ \{x, y\} \in P_2(E),$$

where Φ denotes the cumulative normal distribution function $N(0, 1)$.

Let us now evocate the case of concern where the set E is defined by the set denoted I of the rows of a contingency table that we assume labeled by $I \times K$:

$$\{n_{ik}/(i, k) \in I \times K\}$$

Remind that n_{ik} is the cardinality of the subset of the observation set where the conjunction of the values (i, k) occurs, $(i, k) \in I \times K$. As we have seen, this data structure can be formalized by the simplest case of the system S (see above). Important and minute works have been devoted in our context to this data structure and its generalizations ([Lerman & Tallur, 1980], [Lerman, 1984], [Tallur, 1988]).

The first theoretical aspect of the classification of the row set I, concerns the mathematical representation of I with respect to K. This can be inspired by the correspondence analysis representation [Benzecri, 1973] in which a given row i is represented by a weighted point in an euclidean space, the components of the latter being defined by the empirical distribution of the column variable, namely

$$\{f_k^i = \frac{n_{ik}}{n_{i.}}/k \in K\},$$

where $n_{i.} = \sum\{n_{ik}/k \in K\}$. The assigned weight is then defined by $f_i = n_{i.}/n$, where n indicates the size of the learning set : $n = \sum\{n_{i.}/i \in I\}$, $i \in I$. Moreover, the euclidean representation space is provided by the chi-square metric, which intervenes in the raw simi-

larity index.

A dual representation of E, coded by I, has also been studied. For this, K is represented in an analogous way by a set of weighted geometrical points in R^N ($N = card(I)$). And then, E is interpreted in terms of linear forms, as the case is for numerical attributes describing an object set.

Now reconsider the most general situation for the data structure. We may suppose that whatever the nature of the set E to be classified ($E = A$ $E = O$ (respectively C), according to the notation of the beginning of this section), we have a probabilistic similarity table

$$\{P(x, y)/\{x, y\} \in P_2(E)\}$$

which contains $N(N-1)/2$ values lying in the interval $[0, 1]$. On the other hand it can be established that each element of the preceding table becomes, under the $h.n.l$, a random uniform distributed variable on the interval $[0, 1]$.

4.3 Family of criteria of the "vraisemblance du lien maximal" ("maximal link likelihood")

We have to be able to compare, by means of a similarity (or dissimilarity) index, disjoint subsets of the set E to be classified. Denote σ this similarity index. In these conditions, the algorithm of ascendant hierarchical classification, using σ to build a classification tree on E, is a "trivial" mathematical principle : "At each step, join the class pairs which realize the maximum value of σ". The leaves of the classification tree represent singleton classes, which define each a subset of E with exactly one element. The root of this tree represent the whole set E.

Let $\{G, H\}$ be a class pair ; G and H are disjoint subsets of E. In order to establish the AVL similarity index $\sigma(G, H)$, we start by defining, from topological considerations, a raw similarity index $s(G, H)$. Then, by associating under the $h.n.l$ with $\{G, H\}$ a random pair of disjoint subsets $\{G*, H*\}$, one may measure the degree of exceptionality of the bigness of $s(G, H)$. This measure refers to a probability scale and gives $\sigma(G, H)$.

This idea also appears fundamental in information theory, where the higher the quantization of an event, the more unlikely it is. The events with which we are concerned in the AVL methodology are the observed relations between descriptive variables (respectively, described objects or concepts), or here, between variable (attribute) classes (respectively, object classes or category classes).

In the AVL hierarchical classification method, the probabilistic similarity index between disjoint subsets of the set E to be classified, is established from the above probabilistic si-

milarity table on E.

For the family of criteria of the "vraisemblance du lien maximal" the raw similarity index is defined by

$s(G,H) = max\{P(x,y) \mid (x,y) \in G \times H\}$,

where the notations have been defined above.

To the couple (G,H), we associate a couple $(G*,H*)$ of independent random classes, where $G*$ (respectively $H*$) is composed of independent elements relating to the formal and statistical structures of the G-elements (respectively H-elements). This association corresponds exactly to the h.n.l. mentionned above.

Thus $s(G*,H*)$ becomes a random index. Then, the definitive association index between the two classes G and H takes the following form

$$\begin{aligned} \sigma(G,H) &= Prob\{p(C*,D*) \leq p(C,D)/h.n.l.\} \\ &= [p(G,H)]^{l \times m} \end{aligned}$$

where $l = card(G)$ and $m = card(H)$ [Lerman 1970,1981].

The latter index corresponds to what we call the pure form of the "vraisemblance du lien maximal" ("maximal link likelihood") criterion, that we can denote by

$VL_1(G,H) = [p(G,H)]^{l \times m}$

Following the works of Nicolau and Bacelar-Nicolau ([Nicolau,1981],[Bacelar-Nicolau,1985]), we have suggested the family of criteria :

$VL_\xi(G,H) = [p(G,H)]^{(l \times m)^\xi}$

where ξ is a real parameter between 0 and 1. This family goes from the maximal link ($\xi = 0$) to the pure form of the maximam link likelihood criterion ($\xi = 1$).

For reasons of accuracy in computing, we consider the strictly increasing function

$S_\xi(G,H) = -log_2\{-log_2[VL_\xi(G,H)]\}$

which leads to exactly the same classification tree.

The value adopted here for ξ is 0.5. This is the reason for which the hierarchical classification method was called $AVL_{0.5}$. This value of ξ works very well. It was initially the only value proposed by the previous authors, with the following expression of the criterion

$[p(G,H)]^{\sqrt{l \times m}}$

The latter has some formal justification deduced by analogy. Associate with it, the equivalent dissimilarity index between classes, by applying the function $-log_2$. This criterion becomes

$\sqrt{l \times m}\ min\{-log_2(P(x,y)) \mid (x,y) \in G \times H\}$

In this, $-log_2(P(x,y))$ defines a dissimilarity index between x and y associated with the similarity one $P(x,y)$. It varies from zero (perfect ressemblance) to infinity (complete disemblance).

Consider now a structural situation where it is admissible to mathematically represent E by a cloud of points in an euclidean space. Moreover, assume that the distance d associated with the latter provides an adequate dissimilarity notion on E. Under these conditions, the inertial Ward criterion (1963) can be envisaged. It can be written with the above notations

$\frac{l \times m}{l+m}\ d^2[g(G), g(H)]$,

where $g(X)$ indicates the centroïd of X.

Notice that the multiplicative coefficient $l \times m/(l+m)$ is nothing else, to the factor 2, the harmonic mean between l and m. Therefore the passage from the Ward criterion to the $AVL_{0.5}$ one can be obtained by :

(i) replacing the distance square between the classe centroïds by the lowest dissimilarity $-log_2 P$ between two respective elements of the two classes ;

(ii) replacing the harmonic mean between l and m, by the geometric one.

Now, other values of ξ can be considered in practical applications and the most important of them is $\xi = 1$. Finally, if we have to retain two values for ξ, there are $\xi = 1$ and $\xi = 0.5$.

4.4 Significant levels and significant nodes of a classification tree

A given level of a classification tree on the set E to be classified, provides a partition on E. We may assume this classification tree obtained by a hierarchical clustering method ; for

example $AVL_{0.5}$ (see above). The detection of the significant levels and significant nodes of the classification tree on E is based on the elaboration of an association criterion which matches a given partition $\pi(E)$ and a coded information relative to the ressemblances between the elements of E. This information can have numerical or ordinal natures. In the former case it defines a valuation –given by the similarity coefficient– on the set $F = P_2(E)$ of unordered element pairs of E. And in the latter case, the total preorder on F associated with the similarity index is retained. More precisely by considering the above similarity coefficient P, this total preorder that we denote by $\omega(E)$ is defined as follows :

$$[\forall(p,q) \in F \times F], p \leq q \Leftrightarrow P(p) \leq P(q)$$

We will outline here the case where we deal with $\omega(E)$. Detailed treatment is given in [Lerman 1981,1983],[Lerman & Ghazzali,1991].

For comparing $\omega(E)$ with a partition $\pi = \pi(E)$ (which can be given by a particular classification tree level), we interpret π as inducing a total preorder into two classes on the set $F : S(\pi)$ and $R(\pi)$, where $S(\pi)$ (respectively $R(\pi)$) is the set of separated (respectively joined) pairs. We set

$$S(\pi) < R(\pi)$$

In this way, one finds ourselves faced with the comparison of two combinatoric structures of the same nature (total preorders on F). For this comparison –as expressed previously– we have built an association coeficient which is statistically normalized with respect an h.n.l. and which we denote here by $Q[\omega(E), \pi(E)]$ (see above references). Let us consider now the following two sequences associated with a classification tree on E :

$$\{\sigma_i = Q[\omega(E), \pi_i(E)] \mid 1 \leq i \leq l\}, \{\tau_i = \sigma_i - \sigma_i - 1 \mid 2 \leq i \leq l\}$$

where $\pi_i(E)$ is the emerged partition at the ith level of the classification tree and where l is the total number of levels. Notice that the second sequence is deduced from the first one by considering the increasing rate of the coefficient Q between two consecutive levels. A 'significant' level corresponds to a local maximum of the distribution $\{\sigma_i \mid 1 \leq i \leq l\}$ on the increasing sequence of tree levels. Whereas, a 'significant' node corresponds to a local maximum of the distribution $\{\tau_i | 2 \leq i \leq l\}$. A 'significant' level gives empirically a stable and relevant partition of which the number of classes can be more or less calibrated. On the other hands, a 'significant' node indicates the achievement of a cluster at a given synthesis degree.

5 Conclusion and perspectives

In this work, we have set up a method for extracting few number of relevant predictive binary attributes from caterogical nominal attributes having very large number of values. The aim of this technique is here situated in the context of the construction of a binary decision tree by CART method. However, our binarization technique is independent of the prediction method used. And then, any prediction method based on binary attributes can employ it.

Otherwise, in case where the initial categorical attributes have not too large value sets, we have shown how to form more informative attributes by "cartesian multiplication". This formation precedes the application of the binarization method is which tree-structured attributes are built for prediction purposes.

The ARCADE method (see section 3.1 for its general scheme) gives strategy which enables to treat, for the first time, description by sets of four amino acids taken together in the protein secondary structure prediction problem. By comparison with other methods applied for this problem, our results are very competitive.

We plan to generalize ARCADE method to the case of non-binary decision trees. This shall allow us to introduce a new splitting rule taking into account the ultrametric structures of the classification trees which respectively organize the value sets of the predictive categorical attributes.

References

Almuallim, H., Akiba, Y. & Kaneda, S. (1995) On Handling Tree-Structured Attributes in Decision Tree Learning. *International Conference on Machine Learning*, 1995.

Bacelar-Nicolau, H. (1985): The affinity coefficient in Cluster Analysis. *Methods of Operations Research*, vol.53, 507-512.

Benzecri, J.P. (1973): *L'Analyse des Données.* Tome 2 : *L'Analyse des Correspondances.* Dunod, Paris.

Breiman, L., Friedman, J. H., Olshen, A. and Stone, C. J. (1984): *Classification and Regression Trees (1984).* Wadsworth, Belmont.

Buntine, W. & Niblett, T. (1992): A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8, 75-86, 1992.

Colloc'h, N., Etchebest, C., Thoreau, E., Henrissat, B. & Mornon J.P.: *Protein Engineering (1993)*,6,377-382.

Cost, S. & Salzberg, S. (1993): A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, 10, 57-78.

Costa, J.F.P. (1996): *Coefficients d'association et binarisation par la classification hiérarchique dans les arbres de décision. Application à l'identification de la structure secondaire des protéines.* Thèse de doctorat, Université de Rennes I, Rennes, France, 1996.

Heath, D., Kasif, S. & Salzberg, S. (1993): Induction of Oblique Decision Trees. *Proc. of the IJCAI 93*, 13th International Joint C. On A.I., Chambéry, France, 1993.

Krzanowsky, W.J. (1975): Discrimination and Classification Using Both Binary and Continuous Attributes. *Journal of The American Statistical Association*, Volume 70, Number 352, pp. 7

Lerman, I.C. (1970): Sur l'analyse des données préalable à une classification automatique. *Proposition d'une nouvelle mesure de similarité.* Revue Mathématique et Sciences Humaines, 32, pp.5-15.

Lerman, I.C. (1981): *Classification et analyse ordinale des données.* Paris, Dunod.

Lerman, I.C. (1983): Sur la signification des classes issues d'une classification automatique. *Numerical Taxonomy*, edited by J.Felsenstein, NATO ASI Series Vol. G1, Springer-Verlag, Berlin, pp. 179-198.

Lerman, I.C. (1984): Analyse classificatoire d'une correspondance multiple, typologie et régression. *Data Analysis and Informatics III*, edited by E. Diday, North Holland, pp. 193-221.

Lerman, I.C. (1987): Construction d'un indice de similarité entre objets décrits par des variables d'un type quelconque. Application au problème du consensus en classification. *Revue de Statistique Appliquée*, xxxv(2), pp. 39-76.

Lerman, I.C. (1991): Foundations of the Likelihood Linkage Analysis (LLA) classification method. *Applied Stochastic and Data Analysis*, Vol. 7, pp. 63-76.

Lerman, I.C. (1992a): Conception et analyse de la forme limite d'une famille de coefficients statistiques d'association entre variables relationnelles I. *Revue Mathématique Informatique et Sciences Humaines*, 30è année, n^o 118, Paris, pp. 35-52.

Lerman, I.C. (1992b): Conception et analyse de la forme limite d'une famille de coefficients statistiques d'association entre variables relationnelles II. *Revue Mathématique Informatique et Sciences Humaines*, 30è année, n^o 119, Paris, pp. 75-100.

Lerman, I.C. (1993): Likelihood linkage analysis (LLA) classification method (Around an example treated by hand). *Biochimie 75*, Elsevier éditions, 1993, pp. 379-397.

Lerman, I. C. & Costa, J. F. (1996): Coefficients d'association et variables à très grand nombre de catégories dans les arbres de décision. Application à l'identification de la structure secondaire d'une protéine. *Publication Interne Irisa n^o 982 et Rapport de Recherche Inria n^o 2803, 46 pages.*

Lerman, I.C. & Ghazzali, N. (1991): What do we retain from a classification tree? an experiment in image coding. *Symbolic-Numeric data analysis and learning*, edited by E. Diday and Y. Lechevallier, Nova Science Publishers. Proceedings of the Conference of Versailles, September 18-20,1991, pp. 27-42, 1991.

Lerman, I.C., Peter, Ph. et Leredde, H. (1993-1994): Principes et calculs de la méthode implantée dans le programme CHAVL (Classification Hiérarchique par Analyse de la Vraisemblance des Liens). *La Revue de Modulad*, n^o 12, pp.33-70 et n^o 13, pp.63-90.

Lerman, I.C. & Tallur, B. (1980): Classification des éléments constitutifs d'une juxtaposition de tableaux de contingence. *Revue de Statistique Appliquée*, n^o 28, Paris, pp. 5-28.

Liu, W.Z. & White, A.P. (1994): The importance of Attribute Selection Measures in Decision Tree Induction. *Machine Learning 15: 25-41, 1994.* Kluwer Academic Publishers - Manufactered in the Netherlands, 1994.

Müller, W. & Wysotzki, F. (1994): A Splitting Algorithm, Based on a Statistical Approach in the Decision Tree Algorithm CAL5. *Proc. of the ECML 94*, 7th European Conference on ML, Catania, Sicily, April 1994.

Nakhaeizadeh, G. (1994): Interaction Between Machine Learning and Statistics, An Overview. *Proc. of the ECML 94*, 7th European Conference on ML, Catania, Sicily, April 1994.

Nicolaü, F. (1981): *Criterios de analise classificatoria hierarquica baseados na funçao de distribuiçao.* Ph.D. thesis, Science Faculty of Lisbon.

Qian, N. & Sejnowsky, T. (1988): Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology*, 202, 865-884.

Quinlan, J.R. (1986): Induction of Decision Trees. *Machine Learning*, pp.81-106.

Rost,B. & Sander,C. (1993): Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* 232, pp. 584-599, 1993

Solovyev, V. & Salamov, A. (1994): Predicting α-helix and β-strand segments of globular proteins. *CABIOS.* Vol. 10 n^o 6, pp. 661-669, 1994.

Tallur, B. (1988): *Contribution à l'analyse exploratoire de tableaux de contingence par la classification.* Thèse de Doctorat ès Sciences, Université de Rennes I.

Taylor, C. C., (1994): Distance-based Decision Trees. *Proc. of the ECML 94*, 7th European Conference on ML, Catania, Sicily,

Van de Merckt, T. (1993): Decision Trees in Numerical Attribute Spaces. *Proc. of the IJCAI 93*, 13th International Joint C. On A.I., Chambéry, France.

Ward, J.H. (1963): Hierarchical grouping to optimise an objective function. Journal of American Statistical Association, JASA, 58, pp. 236-244.

Zhang, X., Mesirov, J. & Waltz, D. (1992): A hybrid system for protein secondary structure prediction. *Journal of Molecular Biology*, 25, 1049-1063.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399